(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 102667755 B
(45)授权公告日 2016.08.03

(21)申请号 201080050251.0

(22)申请日 2010.09.03

(30)优先权数据
61/239,745 2009.09.03 US

(85)PCT国际申请进入国家阶段日
2012.05.02

(86)PCT国际申请的申请数据
PCT/US2010/047832 2010.09.03

(87)PCT国际申请的公布数据
WO2011/029015 EN 2011.03.10

(73)专利权人 华莱士·E.·拉里莫尔
地址 美国弗吉尼亚州

(72)发明人 华莱士·E.·拉里莫尔

(74)专利代理机构 上海脱颖律师事务所 31259
代理人 脱颖

(51)Int.Cl.
G06F 17/00(2006.01)
G06F 17/10(2006.01)
G06F 17/16(2006.01)

(56)对比文件
CN 1577339 A,2005.02.09,
US 2005/0021319 A1,2005.01.27,
丁千 等.机翼颤振的非线性动力学和控制研究.《科技导报》.2009,第53-61页.

审查员 王伟

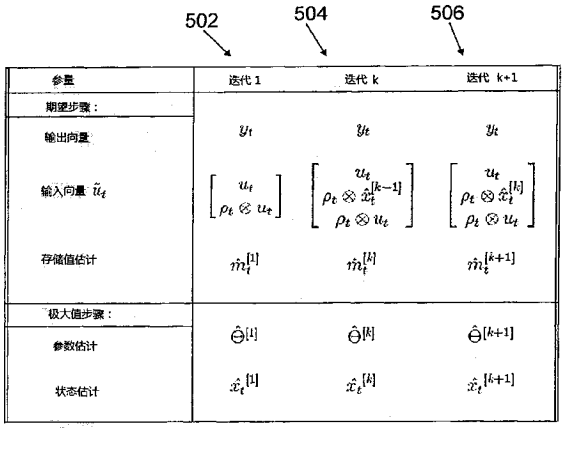
权利要求书1页 说明书15页 附图6页

(54)发明名称

通过迭代线性子空间计算对时变、变参数和非线性系统进行实验建模的方法和系统

(57)摘要

一种用于估计微分或差分方程的方法和系统，该方程可以控制非线性、时变和变参数的动态过程或系统。所述用于估计所述方程方法和系统是通过对所述方程的观测到的输出和(当需要时的)输入数据进行估计来实现的。所述方法和系统可采用任何如下的系统或过程：该系统或过程能够被描述成非线性、时变和变参数的差分方程并且可用于在描述详细的系统或方法性能时自动提取该差分方程，该系统或方法性能用于系统控制、故障检测、状态估计以及系统或方法中有变化时它们的预测和修改。



CN 102667755 B

1.一种通过将子空间识别方法用于具有一般调度函数的线性变参数(LPV)和非线性变参数系统以使飞机机翼颤振稳定的方法,包括:

通过传感器检测飞机机翼颤振;

将检测到的机翼颤振数据存储于存储器中;

对所存储的机翼颤振数据执行增加阶的自回归(ARX)模型拟合,所述拟合包括:计算AIC以确定最优阶,和选择具有最优阶的ARX模型;

处理器执行校正的新值计算,所述校正的新值计算通过使用ARX模型来确定一个或者多个机翼颤振数据新值输入对新值输出的效应,和从新值输出中减去所述效应来确定一个或多个校正后的机翼颤振数据新值输出;

处理器通过执行校正后的新值输出和旧值之间的规范变量分析(CVA)来确定状态估计;

通过AIC来确定状态阶;

将状态估计代入一个或多个状态方程;

对所述一个或多个状态方程执行线性回归运算以确定状态方程的系数矩阵;以及

用具有上述系数矩阵的状态方程的形式提供飞机机翼颤振动态模型,以将子空间识别方法用于具有一般调度函数的LPV和非线性变参数系统。

2.根据权利要求1所述的方法,其中,ARX模型拟合具有假定估计的最大ARX阶的线性回归问题。

通过迭代线性子空间计算对时变、变参数和非线性系统进行实验建模的方法和系统

[0001] 相互参考应用

[0002] 本申请根据35U.S.C.§120主张2009年9月3日提交的申请号为61/239,745的美国临时专利申请的优先权，其内容通过引用整体并入本申请。

背景技术

[0003] 根据测得的输出数据和可能的输入数据来对非线性和时变动态过程或系统进行建模是该技术的新兴领域。根据理论或应用的领域，这在统计学中称为时序分析，在工程学中称为系统识别，在心理学中称为纵向分析，且在财务分析中称为预测。

[0004] 过去，已对子空间系统识别方法进行改革、对具有反馈的系统的最优方法进行大量改进和改良，以及对包括双线性系统和线性变参数(LPV)系统的非线性系统的方法进行研究。子空间方法可以避免不收敛的迭代非线性参数最优化，并且可以对高阶大规模的系统使用具有相当大价值(considerable value)的数值稳定性方法。

[0005] 在时变和非线性系统领域，已经开展了一些工作，虽然没有达到期望的结果。这些工作在最好将线性子空间方法的直接展开(direct extension)用于非线性系统建模方面代表了本领域的现状。这种方法将旧值和新值表示为旧值输入和输出的非线性函数的线性组合。该方法的一个结果是，在测得的输入、输出、状态和所使用的旧值滞后的数量方面，旧值和新值的维数成指数增长。当只使用每种这些变量中的几个时，旧值的维数计算为超过10^4或甚至多于10^6。对于典型的工业过程，旧值的维数很容易超过10^9或甚至10^{12}。这些极大的数字最多导致利用效率低的结果。

[0006] 其它方法使用估计模型中非线性项的迭代子空间方法，因而需要很少的计算。这种方法包括启发式算法，且在具有随机调度函数(即，具有白噪声特性)的LPV系统情况下已经用于高精度模型识别。然而，问题之一是在大部分LPV系统中，调度函数通常由特定应用确定且特征是通常是非随机的。已经实施了一些修改以试图在非随机调度函数的情况下提高精度，结果是，所尝试的修改在充分提高模型精度方面没有获得成功。

[0007] 在更通常的内容中，识别非线性系统的通常问题被认为是通常的非线性规范变量分析(CVA)过程。该问题可用洛伦茨吸引子(Lorenz attractor)，由简单的非线性差分方程描述的混沌非线性系统，来阐述。以此方式，旧值和新值的非线性函数被确定来描述过程的状态，该过程的状态接着被用于描述系统的非线性状态方程。这种方法的一个主要难题是找到可行的计算实施，因为众所周知的是，所需要的旧值和新值的非线性函数的数量是成指数增长的。在寻找一般非线性系统所存在的系统识别问题的解决方案时经常会遇到这个难题。

[0008] 因此，在下面描述的一些示例性的实施例中，描述的方法和系统可以实现相当大的改进且还可以在观测的‘大量采样’是可用的情况下产生最优结果。并且，该方法不是‘特定的(ad hoc)’，而是可以包括最优的统计方法。

发明内容

[0009] 一个示例性的实施例描述了一种用于使用非线性、时变和变参数动态过程的方法。该方法可用于产生减小的具有时变项的系统模型。该方法可包括如下步骤：展开状态差分方程；将差分方程表示为关于输出和增量输入的线性、时不变系统；以及估计状态方程的系数。

[0010] 另一示例性的实施例描述了一种用于估计一组控制非线性、时变和变参数过程的方程的系统。该系统具有第一输入、第二输入、反馈框和时延框。并且，在该系统中，第一输入和第二输入经过反馈框到达时延框，从而产生输出。

具体实施方式

[0011] 在下面的说明和指向本发明特定实施例的相关附图中描述了本发明的一些方面。本领域技术人员将意识到可以设计替代的实施例，只要其不背离权利要求的精神或范围。并且，这里不详细描述本发明示例性的实施例的公知原理，或省略这些公知原理，以免使得本发明的相关详细内容变得模糊不清。

[0012] 如这里所使用的，词组“示例性的”表示“作为例子、实例或例证”。这里描述的实施例不只限于示例性的实施例，而可以包括示例性的实施例以外的实施例。应该理解的是，所描述的实施例不必然被解释为对其他实施例是优选的或具有优势的。并且，术语“本发明的实施例”、“实施例”或“发明”不要求本发明的所有实施例都包括所描述的特征、优势或操作方式。

[0013] 此外，许多实施例是根据将要由例如计算设备的元件执行的操作顺序来描述的。应该认识到的是，可以通过特定电路(例如，特定用途集成电路(ASICs))，通过由一个或多个处理器执行的程序指令，或由这两者的组合来执行这里描述的各种操作。并且，这里描述的这些操作顺序被认为整个包含在任何形式的计算机可读存储媒介内，该存储媒介具有存储在其内的一组对应的计算机指令，当执行这些指令时会使得相关的处理器执行这里描述的功能(functionality)。因此，本发明的各个方面可以通过多种不同的形式具体体现，所有这些形式已经预期落入权利要求的主题范围内。并且，对于这里描述的每个实施例，任何这样的实施例的对应形式都在这里描述为，例如“逻辑被配置来(logic configured to)”执行所描述的操作。

[0014] 通常参考示例性的图1-6，描述了用于对时变、变参数和非线性差分方程进行实验建模的方法和系统。所述方法和系统可以被实施和使用来产生各种结果，并且所述方法和系统可以有效地实施。

[0015] 如示例性的图1所示，显示了根据一个示例性的实施例，对时变、变参数和非线性差分方程进行实验建模的一套方法的流程图。这里，在102处，使用一组时变、变参数和(如果需要的话)非线性状态空间差分方程。然后在104处，关于所选择的一组基函数来展开所述方程，例如非线性输入-输出方程可以以x_t和u_t的多项式展开。然后在106处，例如根据输出y_t和增量(augmented)输入u_t将差分方程表示为线性时不变形式，该线性形式可以包括输入u_t和基函数，例如，输入u_t、调度函数ρ_t和状态x_t的多项式。

[0016] 示例性的图2显示了示例性的流程图，其中使用差分方程的线性、变参数系统。在

该实施例中在202处，使用一组线性、变参数状态空间方程，如下面显示的方程1和方程2。

[0017] $x_{t+1}=A_0x_t+B_0u_t+[A_1\rho_t(1)+\ldots+A_s\rho_t(s)]x_t+[B_1\rho_t(1)+\ldots+B_s\rho_t(s)]u_t$ 方程1

[0018] $y_t=C_0x_t+D_0u_t+[C_1\rho_t(1)+\ldots+C_s\rho_t(s)]x_t+[D_1\rho_t(1)+\ldots+D_s\rho_t(s)]u_t$ 方程2

[0019] 然后在204处，可以用关于调度函数ρ_t、状态x_t和输入u_t的多项式展开状态空间差分方程，例如如下面的方程3和4所示。

[0020] $x_{t+1}=A_0x_t+B_0u_t+[A_1\cdots A_s](\rho_t\otimes x_t)+[B_1\cdots B_s](\rho_t\otimes u_t)$ 方程3

[0021] $y_t=C_0x_t+D_0u_t+[C_1\cdots C_s](\rho_t\otimes x_t)+[D_1\cdots D_s](\rho_t\otimes u_t)$ 方程4

[0022] 接下来，在206处，可以以初始输出y_t和增量输入$[u_t,(\rho_t\otimes x_t),(\rho_t\otimes u_t)]$的形式来表示差分方程，增量输入$[u_t,(\rho_t\otimes x_t),(\rho_t\otimes u_t)]$是关于$u_t$、$x_t$和$\rho_t$的函数。差分方程具有可被估计的线性非时变的未知系数(A_0,$[B_0\ A_\rho\ B_\rho]$,C_0,$[D_0\ C_\rho\ D_\rho]$)，差分方程如下面显示的方程5和6。

[0023] $$x_{t+1}=A_0x_t+[B_0A_\rho B_\rho]\begin{bmatrix}u_t\\ \rho_t\otimes x_t\\ \rho_t\otimes u_t\end{bmatrix}$$ 方程5

[0024] $$y_t=C_0x_t+[D_0C_\rho D_\rho]\begin{bmatrix}u_t\\ \rho_t\otimes x_t\\ \rho_t\otimes u_t\end{bmatrix}$$ 方程6

[0025] 然后，如在208处所示，增量输入$[u_t,(\rho_t\otimes x_t),(\rho_t\otimes u_t)]$和(在一些示例性的实施例中特别是)$\rho_t\otimes x_t$，可以包含未知的状态向量$x_t$，所以可以采用或需要迭代。因此，在这些示例性的实施例中，正如下面更详细描述的，可以采用使用迭代算法的迭代。

[0026] 示例性的图3显示了可被执行用于迭代子空间识别的迭代算法的流程图。在这个实施例中，如在302处看到的，非线性差分方程可以用增加的(additive)基函数来展开，并且可以用增量输入u_t的线性时不变形式来表示该非线性差分方程。在一些例子中，这可以包括包含输出、状态和调度函数的非线性基函数。

[0027] 然后，在304处，状态估计$\hat{x}_t^{[0]}$是未知的，其等于不在LPV模型中的(A_ρ C_ρ)项，因此对应的项可以从一组增量输入$\tilde{u}_t$中删除。然后，该迭代算法可以将增量输入用作输入来执行并且可以计算模型参数的估计$\hat{\Theta}^{[1]}$。然后，例如对估计的参数值$\hat{\Theta}^{[1]}$使用卡尔曼滤波器，根据一步预估的新方法(innovations)可以计算出状态估计$\hat{x}_t^{[1]}$。然后，可以估计似然函数。

[0028] 接下来，在306，可以执行迭代k(k>2)。这里，状态估计$\hat{x}_t^{[k-1]}$对于所有t都是初始化的。该迭代算法可以将增量输入用作输入来执行并且可以计算参数的估计$\hat{\Theta}^{[k]}$。再次，例如，对估计的参数值$\hat{\Theta}^{[k]}$使用卡尔曼滤波器，则可以计算出状态估计。同样可以执行一步预估的新方法并且同样可以估计似然函数$\rho\left(Y_{1:N}\middle|\hat{\theta}_{1:N};\hat{\Theta}^{[k]}\right)$。

[0029] 在308处可以验证收敛性。这里，可以比较log似然函数的值的变化以及迭代k-1和迭代k之间的状态阶(orders)。如果，在一些示例中，状态阶相同并且log似然函数的变化小

于所选择的阈值ε。该阈值在一些示例中小于1，例如为0.01，然后迭代结束或停止。否则，似然函数变化值高于所选择的阈值ε，则回到上面的步骤306并且执行迭代k+1。接下来执行迭代k+1，然后再次验证收敛性。

[0030] 在另一个实施例中，对于使用例如线性非时变系统所采用的子空间方法的系统中自相关的误差和反馈的情况，可以使用不同的方法直接和简单地获得未知参数的最优或期望的估计。这可以通过用不同的形式表述这个问题来实现，这些不同的形式导致需要在状态估计上进行迭代；然而，迭代的次数可能是非常低的，并且，为了进一步简化系统和其展开式(development)，可以消除随机噪声。

[0031] 例如，考虑到线性系统，其中系统矩阵是如以下方程所示形式的调度参数向量$\rho_t=(\rho_t(1)\ \rho_t(2)\ \ldots\ \rho_t(s))^T$的时变函数：

[0032] $x_{t+1}=A(\rho_t)x_t+B(\rho_t)u_t$ 方程7

[0033] $y_t=C(\rho_t)x_t+D(\rho_t)u_t$ 方程8

[0034] 对于调度参数的仿射相关性(affine dependence)，状态空间矩阵可具有以下方程9至12的形式。

[0035] $$A(\rho_t)=A_0+\sum_{i=1}^{s}\rho_t(i)A_i$$ 方程9

[0036] $$B(\rho_t)=B_0+\sum_{i=1}^{s}\rho_t(i)B_i$$ 方程10

[0037] $$C(\rho_t)=C_0+\sum_{i=1}^{s}\rho_t(i)C_i$$ 方程11

[0038] $$D(\rho_t)=D_0+\sum_{i=1}^{s}\rho_t(i)D_i$$ 方程12

[0039] 在以上方程中，应该注意到，左边的矩阵表示为右边由调度参数ρ_t表示的线性组合$(1\rho_t(1)\ \rho_t(2)\ \ldots\ \rho_t(s))^T$，该线性组合被称为仿射形式。在进一步的论述中，该仿射形式的符号将被用于系统矩阵$A=[A_0\ A_1\ \ldots\ A_s]=[A_0\ A_\rho]$，且类似地可用于B、C和D。

[0040] 在一些其它示例性的实施例中，LPV系统类的系统识别方法具有多种潜在的应用和经济价值。这样的系统可以包括，但不限于空气动力学的和流体动力学的交通工具(例如飞机和轮船)、机动引擎动力学、涡轮引擎动力学、化学处理(例如搅拌槽反应器和蒸馏塔)和其他系统。一个特点是，系统动力学在任何已知的操作点ρ_t都可被描述为线性系统。调度参数ρ_t是关于操作点变量(例如，但不限于速度、压力、温度、燃料流和类似的变量)的复杂的非线性函数，该操作点变量可以是用于在可能未知的常数矩阵A、B、C和D范围内表征该系统动态的已知的或精确测得的变量。还可假设ρ_t是可计算的或可从任何这样的操作点变量的信息中确定。例如，机动引擎的LPV模型可以包括LPV状态空间方程，该LPV状态空间方程明确地将向量ρ_t项表示为关于各种操作点变量的极其复杂的非线性函数。在这里描述的一些示例性的实施例中，当执行系统识别计算时，仅期望的是，调度参数ρ_t是可用的。这样放宽了实时使用或如实时控制或滤波这样的应用的要求。

[0041] 为了简化该论述，可以通过使调度参数ρ_t与输入u_t、状态x_t相关联将LPV方程写成时不变形式，如

[0042] $$x_{t+1}=A_0x_t+B_0u_t+[A_1\cdots A_s](\rho_t\otimes x_t)+[B_1\cdots B_s](\rho_t\otimes u_t)$$ 方程13

[0043] $y_t=C_0x_t+D_0u_t+[C_1\cdots C_s](\rho_t\otimes x_t)+[D_1\cdots D_s](\rho_t\otimes u_t)$ 方程14

[0044] 这里，⊗可以表示克罗内克积$M\otimes N$，对于由i、j块形成的作为隔开矩阵的任何矩阵M和N，$M\otimes N$被限定为$(M\otimes N)_{ij}=m_{ij}N$，其中M的i、j项表示为$m_{ij}$。并且，符号[M;N]=$[M^T\ N^T]^T$还被使用来堆栈(stacking)向量或矩阵M和N。则上述方程13和14还可以写成如下所示的方程15和16的格式。

[0045] $$x_{t+1}=A_0x_t+[B_0A_\rho B_\rho]\begin{bmatrix}u_t\\ \rho_t\otimes x_t\\ \rho_t\otimes u_t\end{bmatrix}$$ 方程15

[0046] $$y_t=C_0x_t+[D_0C_\rho D_\rho]\begin{bmatrix}u_t\\ \rho_t\otimes x_t\\ \rho_t\otimes u_t\end{bmatrix}$$ 方程16

[0047] 如下面更详细的论述，上述方程可以表示为具有非线性反馈$f_t=[\rho_t\otimes x_t;\rho_t\otimes u_t]$的线性时不变(LTI)系统，其中状态$x_t$和输入$u_t$乘以时变调度参数$\rho_t$。现在反馈$f_t$输入可被认为是该LTI系统的实际输入。如下面进一步详细的论述，LPV系统描述的矩阵[A_ρ B_ρ;C_ρ D_ρ]是描述LPV系统的LTI反馈表示的恰当量。

[0048] 并且，上述方程现在可表示为如下面所示的方程17和18。

[0049] $x_{t+1}=\tilde{A}x_t+\tilde{B}\tilde{u}_t$ 方程17

[0050] $y_t=\tilde{C}x_t+\tilde{D}\tilde{u}_t$ 方程18

[0051] 因此，对于测量输出$\tilde{y}_t=y_t$和输入$\tilde{u}_t=\left[u_t^T(\rho_t\otimes x_t)^T(\rho_t\otimes u_t)^T\right]^T$，时不变矩阵可以分别为$(\tilde{A},\tilde{B},\tilde{C},\tilde{D})=\left(A_0,\begin{bmatrix}B_0 & A_\rho & B_\rho\end{bmatrix},C_0,\begin{bmatrix}B_0 & C_\rho & D_\rho\end{bmatrix}\right)$。并且，在$\rho_t\otimes x_t$中的$x_t$不是已知或测得的量的情况下，先前的估计$x_t$是可用的或可以被使用，并且可以使用迭代来获得更精确的或期望的估计x_t。

[0052] 在更进一步的示例性的实施例中，LPV系统可表示为具有非线性内反馈的线性时不变系统，该反馈可以包括已知的变参数函数ρ_t(见《非线性系统识别》部分2.1:状态空间法，由Vincent Verdult著，荷兰特文特大学2002年博士论文，其内容通过引用整体并入本申请中)。在该示例性的实施例中，对i($1\le i\le s$)使用奇异值分解将具有秩r_i的系统矩阵P_i分解为如方程19所示。

[0053] $$P_i=\begin{bmatrix}A_i & B_i\\ C_i & D_i\end{bmatrix}=\begin{bmatrix}B_{f,i}\\ D_{f,i}\end{bmatrix}\begin{bmatrix}C_{z,i} & D_{z,i}\end{bmatrix}$$ 方程19

[0054] 则参量可被定义如下，如方程20-23所示。

[0055] $B_f=[B_{f,1}\ B_{f,2}\ \cdots\ B_{f,s}]$ 方程20

[0056] $D_f=[D_{f,1}\ D_{f,2}\ \cdots\ D_{f,s}]$ 方程21

[0057] $C_z^T=\left[C_{z,1}^TC_{z,2}^T\cdots C_{z,s}^T\right]$ 方程22

[0058] $D_z^T=\left[D_{z,1}^TD_{z,2}^T\cdots D_{z,s}^T\right]$ 方程23

[0059] 接下来，LTI系统$P_0=[A_0B_0;C_0D_0]$中的内反馈被假设具有输出$z_t=C_zx_t+D_zu_t$和将z_t输入其中的非线性反馈$f_t=\rho_t\otimes z_t$，该非线性反馈通过矩阵B_f和D_f输入到LTI系统P_0状态方程。反馈系统的x_{t+1}和y_t的状态方程可以如下面的方程24至27所示。

[0060] $x_t=A_0x_t+B_0u_t+B_ff_t$ 方程24

[0061] $y_t=C_0x_t+D_0u_t+D_ff_t$ 方程25

[0062] $z_t=C_zx_t+D_zu_t$ 方程26

[0063] $f_t=[\rho_t\otimes z_t]$ 方程27

[0064] 关于上述现在来参考示例性的图4，如图4中显示的反馈结构以及方程26中，如果假设对于任何时刻t反馈f_t不会对输出z_t产生任何效应，则线性分式变换(LFT)技术中不存在参数相关性(见K.Zhou，J.Doyle和K.Glover(1996)著，普伦蒂斯霍尔出版社出版的《鲁棒与最优控制》，部分10.2，其内容通过引用整体结合在本文中)。如图4中所示，存在具有两个方框(404和402)的系统400，其中方框404可以是较少存储容量的非线性反馈系统，而方框402可以是线性时不变系统。因此，在一些示例性的实施例中，参数相关性可变为仿射的。

[0065] 示例性的图4可以是方程15和16的示意图。状态方程15涉及402中的上部的方框，而测量方程16涉及402中的下部的方框。ΔT 422是采样持续的延时，其中方程15的右边在444处进入422，并且在离开422时方程15的左边等于x_{t+1}。这是与方程14和15类似的递推公式，从而对于该图，时间指标在下一迭代开始之前从“t”变为“t+1”，继续迭代直到进入方框420、430和410。调度参数ρ_t406、输入u_t408和输出y_t446是变量。上部的四个方框表示从左到右分别乘以B_0 418、A_0 420、B_ρ 414和A_ρ 416。类似地，用C代替A和用D代替B，下部的四个方框表示分别乘以D_ρ 424、C_ρ 426、D_0 428和C_0 430。在反馈方框404中，ρ_t与x_t的克罗内克积以及ρ_t与u_t的克罗内克积分别在410和412中形成。在垂直方向对准的方框对从左到右分别乘以变量x_t、u_t、$\rho_t\otimes x_t$和$\rho_t\otimes u_t$。另外，所示的压线的箭头表示加法。并且，ΔT 422表示持续时间的延时块ΔT，对于该示例性的实施例，该延时类似日界线(a date line)那样作用。因此，示例性的图4中的箭头表示操作的时间流程(a time flow)或实际顺序，该时间流程在406、408以及422的输出处开始，继续进行执行该示意图。当到达ΔT 422时，已经执行了抽样t的所有操作，并且当通过ΔT 422时，抽样时间t+1开始。因此，例如，当离开ΔT 422时，保持同样的量，但是所有的时间标志都变为t+1，且贯穿示例性的图4所示的进程。

[0066] 并且，如下面所示，这等价于方程15和16所示的LPV形式，其中状态方程关于调度参数向量p是线性的。

[0067] 在进一步示例性的实施例中且现在使用定义为$f_t=\rho_t\otimes z$的反馈定义(如方程26和27所指出的)，x_{t+1}和y_t的状态方程如下面的方程28和29所示。

[0068] $$x_{t+1}=A_0x_t+B_0u_t+\sum_{i=1}^{s}B_{f,i}\rho_t(i)C_{z,i}x_t+\sum_{i=1}^{s}B_{f,i}\rho_t(i)D_{z,i}u_k$$ 方程28

[0069] $$y_t=C_0x_t+D_0u_t+\sum_{i=1}^{s}D_{f,i}\rho_t(i)C_{z,i}x_t+\sum_{i=1}^{s}D_{f,i}\rho_t(i)D_{z,i}u_k$$ 方程29

[0070] 然后，如果方程19被用于定义上述因子，则方程28和29与方程15和16相同。

[0071] 接下来，在进一步示例性的实施例中，当P_i的秩不是已知的，则输出z_t被设置为状

态和输入$z_t=\left[x_t^T;u_t^T\right]^T$，接着通过静态的非线性被反馈，从而有$f_t=\left[\rho_t\otimes x_t;\rho_t\otimes u_t\right]$。然后，$[C_{z,i}\ D_{z,i}]=[I_{dim\ x}\ 0;0\ I_{dim\ u}]$以及方程30和31如下设置。

[0072] $$P_i=\begin{bmatrix}A_i & B_i\\ C_i & D_i\end{bmatrix}=\begin{bmatrix}B_{f,i}\\ D_{f,i}\end{bmatrix}\begin{bmatrix}C_{z,i} & D_{z,i}\end{bmatrix}$$ 方程30

[0073] $$=\begin{bmatrix}B_{f,i}\\ D_{f,i}\end{bmatrix}\begin{bmatrix}I_{\dim x} & 0\\ 0 & I_{\dim x}\end{bmatrix}=\begin{bmatrix}B_{f,i}\\ D_{f,i}\end{bmatrix}$$ 方程31

[0074] 因此，从上述可知，LPV系数矩阵$P_i=[A_i\ B_i;C_i\ D_i]$是左边状态方程变量$(x_{t+1};y_t)$对非线性项的向量$[\rho_{i,t}\ x_t;\rho_{i,t}\ u_t]$的回归矩阵。通常，LPV系数矩阵$P_\rho=[A_\rho\ B_\rho;C_\rho\ D_\rho]=[B_f;D_f]$是左边状态方程变量$(x_{t+1};y_t)$对非线性反馈项的向量$\left(\rho_t\otimes x_t;\rho_t\otimes u_t\right)$的回归矩阵。

[0075] LTI非线性反馈表述可以解决将现有的子空间识别算法应用到LPV系统的识别时产生的主要障碍，并且克服了先前的关于其它方法中非线性项的数量成指数增长的问题。例如，上述LTI非线性反馈表述可以清楚表明：非线性项$\left(\rho_t\otimes x_t;\rho_t\otimes u_t\right)$可以理解为LTI非线性反馈系统的输入。因此，使用线性子空间方法来直接估计LTI系统状态空间方程的矩阵是可能的，这种估计对于具有输入和反馈的过程是精确的。该方法可直接包括LTI非线性反馈系统的输出y_t和增量输入$\left[u_t;\left(\rho_t\otimes x_t;\rho_t\otimes u_t\right)\right]$的使用。

[0076] 在另一示例性的实施例中，可以在减少了非线性反馈系统的LTI子系统之后确定LTI系统矩阵和状态向量，该非线性反馈系统包括已知的调度函数和LTI子系统的状态。该实施例可以包括对LTI系统状态以及描述该LTI系统的LTI状态空间矩阵进行迭代测量。

[0077] 一个例子可以是将多项式系统认为是关于x和u的线性系统，该系统还具有另外增加的更高阶积的项的输入项，那么，另外的输入是$\rho_t\otimes x_t$和$\rho_t\otimes u_t$。假设调度变量ρ_t实时是可用的，如操作点或测量的变量。如果状态xt的精确估计也是可用的，那么，问题就只是对系统识别直接应用迭代算法。由于变量x_t在解决了系统识别之后才是可用的，因此，可以使用不同的方法。

[0078] 因此，在示例性的第一步骤中，对状态向量进行初始估计。这里，对状态方程中的项，包括变量x_t、u_t和$\rho_t\otimes u_t$但不包括变量$\rho_t\otimes x_t$，进行系统识别。由此，可以获得系统线性时不变(LTI)部分的近似值，该系统给出A_0、B_0、C_0、D_0、B_ρ和D_ρ的估计以及状态向量$X_{1,N}[1]=\left[x_1^T x_2^T\cdots x_N^T\right]^T$的估计。

[0079] 然后，在示例性的第二步骤中，对状态向量进行迭代估计。这里，状态向量$X_{1,N}^{[1]}$可用作方程15和16中的项$\rho_t\otimes x_t$中x_t的初始估计。然后，应用迭代算法来获得系统矩阵A、B、C和D的估计以及$X_{1,N}^{[2]}$的精确估计(refined estimate)。并且，接着重复该步骤直到达到期望的收敛。

[0080] 因而，上述示例性的步骤1至3只进行几次迭代。因此，以示例性的方式，迭代算法被用于解决先前公知的关于LPV系统识别的问题。因此，接下来是使用迭代算法分别直接确定非线性差分方程函数$f(x_t,u_t,v_t)$和$h(x_t,u_t,v_t)$的增加的多项式展开的系数F_{ij}和H_{ij}的示例性论述。对于这样的非线性系统来说，这是非常简洁和精简的参数。因而，可以使用这里

描述的线性时不变系统的迭代算法，其在计算要求方面只具有很少的增加。并且，示例性使用该迭代算法来直接处理状态方程中增加的非线性项，该状态方程包括作为另外的输入的状态向量，如$\rho_t \otimes x_t$。由于状态x_t不是初始已知的，因此这可有助于进行迭代来估计只以系统的线性时不变(LTI)部分A_0、B_0、C_0和D_0开始的状态序列。

[0081] 如方程13和14之后的上述示例性论述，关于调度变量ρ_t是仿射的线性变参数系统可以用包括另外的输入变量$\rho_t \otimes x_t$和$\rho_t \otimes u_t$的时不变形式表示。注意到的是，这包括关于x_t和u_t的非线性函数ρ_t。动态系统在变量的这些非线性函数中是线性时不变的。

[0082] 在进一步示例性的实施例中，可以通过迭代算法来追踪另外的输入产生的效应(effect)。下面概述的示例性的步骤可进一步反应在示例性的图5所示的表格以及示例性的图6的流程图中。

[0083] 使用示例性的图5中的单元502且在示例性的图6的步骤602中，拟合增加阶的ARX模型以观察关于输出y_t和输入$\left[u_t^T(\rho_t \otimes \hat{x}_t^{[k-1]})^T(\rho_t \otimes u_t)^T\right]^T$的迭代k，其中对于k＝1，项$\rho_t \otimes \hat{x}_t^{[k-1]}$不存在输入中且进一步可等于从LPV模型结构中移除关于(A_ρ C_ρ)的项。并且，在图6的步骤602中，计算AIC来确定最优阶。ARX模型拟合可具有线性回归(linear regression)问题：除估计的最大ARX阶以外，该问题不对ARX阶做先前的假设。如果确定的ARX阶接近估计的最大值，则估计的最大ARX阶被加倍且再次计算该回归。

[0084] 在示例性的步骤604中，可以计算校正的新值。可以使用ARX模型来确定新值输入对新值输出产生的效应，且从输出中减去该效应。该效应可以计算新值输出，如果在新值时刻t(in the future of time t)不存在输入，则可从在时刻t的状态获得该新值输出。

[0085] 在示例性的步骤606中，可以执行或计算规范变量分析(CVA)。这里，可以计算旧值和校正的新值之间的CVA。此外，还可计算旧值和校正新值之间的协方差矩阵。这类似于对联合的(joint)旧值-新值协方差矩阵进行SVD以获得ARX模型，旧值-新值协方差矩阵具有旧值协方差的阶。该步骤的结果是获得所谓‘存储值(memory)’的系统状态的估计。

[0086] 在示例性的步骤608中，使用状态方程来执行回归。步骤606中的‘存储值’好像数据一样可用于状态方程中，并且可以获得状态空间矩阵和噪声过程的协方差矩阵产生的估计。这些估计是状态空间模型的渐近ML的估计，而未对ARX或SS模型的参数值进行先前的假设。

[0087] 因此，使用上面所描述的方法，可基于迭代k(如示例性的图5的504所示)中假设的输出和输入，在一次计算中获得迭代算法中的ML解，在迭代k中不对假设的参数值进行迭代。该迭代是非线性项$(\rho_t \otimes \hat{x}_t^{[k-1]})^T$中状态估计的精确结果，该项是在迭代k(504)中假设的数据的一部分。

[0088] 回来参考步骤602，ARX模型拟合，增量输出的维数从dimu增加到dimua＝dimu+dimp(dimx+dimu)。为了拟合ARX模型，由于非线性输入项并且根据存在于输出和增量输入变量之中的统计学显著性动态，确定的ARX阶lagp实质上更高。计算包括对维数为dim ua*lagp的数据协方差矩阵计算SVD，其中lagp是估计的(considered)最大ARX阶。用于SVD的计算具有阶$60*(\text{dim ua}*\text{lagp})^3$，因此，该计算正比于$(\text{dimp}(\text{dimx}+\text{dimu})/\text{dimu})^3$增加。

[0089] 因此，系统输入增加非线性项$\rho_t \otimes x_t$和$\rho_t \otimes u_t$的一个结果是使旧值增长了因子

dimp(dimx+dimu)/dimu，且使计算增长了该因子的立方倍。根据u_t、x_t和ρ_t的特定维数，这是非常重要的，然而，在项的数量或计算时间上不成指数增长。本发明的LPV子空间算法仍然对应于线性时不变系统的子空间系统识别，并且此外，由于状态估计X_t包含的项$\left[u_t^T(\rho_t \otimes x_t)^T(\rho_t \otimes u_t)^T\right]^T$的非线性，对系统状态的估计进行迭代直到收敛是可期望的或者(在一些选项中)所需求的。

[0090] 另一需要考虑的因子是，示例性的步骤606中CVA的结果是状态序列x_t的估计的计算值，表示为m_t。符号'm'用作单词'存储值'来使其区别于实际状态向量和如下面讨论的EM算法中所使用的状态向量的各种估计。结合EM算法的极大值步骤的估计$\hat{m}_t$可被示出为线性时不变系统情况下的系统识别问题的最大似然解。在这种情况下，只使用该算法的一次迭代就可获得全局(global)ML解。这与EM和梯度搜索算法不同，这些算法基本上总是使用许多次迭代。CVA估计m_t实际上是系统状态序列的估计，该系统具有等于LTI系统和较大采样大小(large sample size)情况下最大似然估计的参数。这与如下通常的概念不同：首先获得ML参数估计，然后使用卡尔曼滤波器来估计该状态。该算法在一次迭代中不仅获得了ML参数的最优状态序列，而且在该过程中还确定了最优状态阶。在通常的ML方法中，期望的是或(在一些选项中)需要的是获得状态阶的每种选择的ML估计，然后继续进行假设测试以确定最优状态阶。

[0091] 在另一示例性的实施例中，描述了LPV情况的迭代算法的收敛性。在该实施例中，对非线性差分方程的其它形式采用基本类似的方法。

[0092] 对于任何迭代算法，通常会产生两个问题：(1)该算法是否总是收敛；和(2)以什么速率收敛。在估计的计算实例中，对估计的状态$\hat{x}_t^{[k]}$进行的迭代k可以非常快，例如六个步骤。因此，这里显示的是，假设LPV系统是稳定的情况下，迭代算法与显示为总是收敛的EM算法类(the class of)紧密相关。并且，收敛的速率被计算成是几何学的。后者的结果是唯一的，因为EM算法通常在前面进程非常快但在结束时非常慢。将更详细地描述LPV算法的快速末端收敛(terminal convergence)的原因。下面将详细描述初始化、稳定性和收敛的问题。

[0093] 另外，可在EM算法的背景下论述这里所描述的方法和系统，因为这两者之间存在一些平行性(parallelism)。为了显示LPV算法的收敛性，与适于LPV算法的各种变形一起来讨论Gibson和Ninness中的推导，下面表示为GN且通过引用并入本申请，(S.Gibson和B.Ninness，"多变量动态系统的鲁棒极大似然估计"，自动化，第41卷，第5号，第1667-1682页，2005)。Gibson和Ninness(2005)中的所有方程的数目将包括文件GN中的数量之后的GN，例如(23GN)。

[0094] 在该示例性的实施例中，对GN公式进行如下替换以获得LPV算法：用LPV状态方程替换LTI状态方程，以及对于丢失的数据，用迭代算法中的'存储值'向量m_t替换卡尔曼平滑中的状态估计。这个结果是很重要的，因为对于线性系统，当在GN中，这里所描述的迭代算法可以在较大的采样中用一个步骤获得全局ML参数估计。另一方面，对于线性系统，EM算法需要进行多次迭代来获得ML解。

[0095] 用LPV模型替换LTI模型就是用LPV方程(15)和(16)替换状态空间模型方程11GN。这会产生定理3.1的方程的多种变形，因为GN中的u_t和(A，B，C，D)分别用

$\left[u_t^T(\rho_t \otimes x_t)^T(\rho_t \otimes u_t)^T\right]^T$ 和 $\left(A_0, \left[B_0 A_\rho B_\rho\right], C_0, \left[B_0 C_\rho D_\rho\right]\right)$ 替换。因此‘数据’包括向量 $\rho_t \otimes x_t$，其中状态向量 x_t 不是可用的。

[0096] 为了执行示例性的图5中504所示的迭代k，可以使用关于EM算法的最直接的方法(straight forward approach)。‘丢失的数据’包括作为丢失数据的状态向量xt。(见S.Gibson,A.Wills和B.Ninness(2005),“双线性系统的极大似然参数估计”,IEEE Trans.自动控制,第50卷,第10号,第1581-2005页,其内容通过引用整体并入本申请)。然后,我们可以继续讨论双线性系统,该双线性系统包括与LPV算法的 $\rho_t \otimes x_t$ 完全类似的项 $u_t \otimes x_t$。但是这种EM方法有时会导致在接近最大值时EM算法通常慢收敛的现象。

[0097] 因此,相反,可以使用LPV算法,且该算法会产生快速收敛。因此,在这个示例性的实施例中,子空间方法不是将‘丢失的数据’指定为卡尔曼平滑的估计,而是将CVA状态估计 m_t 或‘存储值’指定为丢失的数据。该存储值mt是通过校正新值和旧值之间的规范变量分析所获得的状态向量的估计,校正新值和旧值是在迭代算法的示例性的步骤606中使用图5中描述的输入和输出向量来获得的。这类似于对与迭代k处输出和输入数据相关的全局ML参数估计进行卡尔曼滤波器状态估计,而不是对最后估计的参数值进行卡尔曼平滑状态估计。这会产生一种区别,因为CVA方法根据基于旧值的校正新值来表示似然函数,从而存储值估计mt只取决于输出和输入数据,且它们的分配取决于与用于迭代k中所使用的输出和输入数据相关的ML估计,而不是该状态的平滑估计。(26GN)的实际条件似然函数 $p_0(\xi_t/z_t)$ 是包括在所有的EM计算中的条件似然函数,其中 $\xi_t^T=\left[x_{t+1}^T \quad y_t^T\right]$ 且 $z_t^T=\left[x_t^T \quad u_t^T\right]$。这是与GN的定理3.3中的似然函数相同的似然函数,该似然函数包括在估计状态空间方程的参数的CVA算法的示例性的步骤608中。区别在于在CVA算法的示例性的步骤608中,期望是关于与迭代k处输出和输入数据相关的真实的全局ML估计,而GN估计是关于在先前的迭代中获得的参数值的期望。

[0098] 并且,在该算法的期望步骤中,使用LPV模型和将存储值 m_t 选作丢失的数据会产生GN中的如下结果。不需要修改部分2,即GN的最大期望(EM)算法中的基础理论。在GN的部分3中,丢失的数据被认为是基于图5的504中迭代k处输入和输出量的CVA状态估计 $\hat{m}_t^{[k]}$。因此,方程(22GN)至(28GN)中的 x_t 可以用 $\hat{m}_t^{[k]}$ 替换。

[0099] 因此,保留GN的定理3.1,但是还可在一个步骤中实现与示例性的图5的输入-输出向量相关的全局ML估计。GN的定理3.2可以用迭代算法替换以获得存储值估计 $\hat{m}_t^{[k]}$。GN的定理3.3是与在该迭代算法中获得的结果相同的结果。另一个步骤可用于根据估计 $\Theta^{[k]}$ 和LPV状态方程给出的线性时变状态方程来计算 $\hat{x}_t^{[k]}$。该步骤是期望获得用于开始下一迭代 k_1 的状态估计 $\hat{x}_t^{[k]}$。该步骤可进一步提供迭代算法以产生 $Q(\theta,\theta') \geq Q(\theta',\theta')$,当且仅当 $\hat{x}_t^{[k]}=\hat{x}_t^{[k+1]}$ 时取等号。

[0100] 然后,原理上,存储值 $\hat{m}_t^{[k]}$ 可用来替换状态估计 $\hat{x}_t^{[k]}$。实际上,在方程(43GN)和(44GN)中状态方程的递归结构上输入(projected)的 $\hat{m}_t^{[k]}$ 可以产生渐近ML状态估计和对应

的最优滤波状态估计 $\hat{x}_t^{[k]}$。因此，在计算 $\rho_t \otimes \hat{x}_t^{[k]}$ 时将 $\square_t^{[k]}$（而不是 $\hat{x}_t^{[k]}$）用作‘输入数据’的一部分会产生基本相同的结果，除了结果中的一些‘噪声’外。但是可计算的噪声可以通过 $\hat{x}_t^{[k]}$ 的另外的小的计算来避免。

[0101] 并且，在一些示例性的实施例中，不需要对这里描述的迭代算法进行‘鲁棒’改进，因为使用主要奇异值分解计算已被推导为是稳健的(robust)并且十多年来也被证实为是如此。如果病态(ill-conditioned)的LPV动态系统将被解决以实现高精度，可能的例外是滤波的状态估计 $\hat{x}_t^{[k]}$ 的计算，该计算可以使用Bierman的平方根方法(G.J.Bierman，离散序列估计的因子分解方法，学术出版社(1977)；Dover再版，纽约(2006)，其内容通过引用整体并入本申请)来执行。

[0102] 在另一示例性的实施例中，可证实的是，LPV系统识别算法可以在接近极大似然解处以几何速率收敛。这里，可以推导出线性系统的结果。LPV系统可以使用同样的方法，但是下面的描述是取决于时间的且具有更大的复杂性。

[0103] 并且，为了简化这里的推导，方程17和18中的时不变反馈可以通过用 $(A,B,C,D,u_t,\tilde{u}_t)$ 替换 $(\square,\square,\square,\square,\tilde{u}_t,u_t)$ 来估计。因此，具有噪声 v_t 的方程17和18用新形式写成如下面的方程32和33所示。

[0104] $x_{t+1}=Ax_t+Bu_t+Kv_t$ 方程32

[0105] $y_t=Cx_t+Du_t+v_t$ 方程33

[0106] 因此，从上述内容可以看出，在测量输出 y_t 和输入 $u_t=\left[\tilde{u}_t^T(\rho_t\otimes x_t)^T(\rho_t\otimes u_t)^T\right]^T$ 时，时不变矩阵分别为 $(A,B,C,D)=(A_0,[B_0,A_\rho,B_\rho],C_0,[B_0,C_\rho,D_\rho])$。然后，求解方程33中的 v_t 并将其代入方程32中，则产生如下方程34。

[0107] $x_{t+1}=Ax_t+Bu_t+K(y_t-Cx_t+Du_t)=(A-KC)x_t+(B-KD)u_t+Ky_t$ 方程34

[0108] 接下来，通过递归将 x_t 代入方程34的右边，可以产生如下方程35。

[0109] $$x_t=\sum_{i=1}^{\infty}(A-KC)^{i-1}\left[(B-KD)u_{t-i}+Ky_{t-i}\right]=Jp_t\left(y_t,\left[\tilde{u}_t(\rho_t\otimes x_t),(\rho_t\otimes\tilde{u}_t)\right]\right)$$ 方程35

[0110] 在方程35中，J包括ARX系数，而 ρ_t 是旧值输出 y_t 并且输入 $u_t=\left[\tilde{u}_t(\rho_t\otimes x_t),(\rho_t\otimes\tilde{u}_t)\right]$。这里，$\tilde{u}_t$ 是初始输入，从而 u_t 包括非线性克罗内克积的项。

[0111] 在进一步示例性的实施例中，渐近地对于较大的采样，进行足够大的迭代k时，J任意地接近常数，从而

[0112] $$\Delta x_t^{[k]}=x_t^{[k+1]}-x_t^{[k]}=Jp_t\left[0,0,\left(\rho_r\otimes\left(x_t^{[k]}-x_t^{[k-1]}\right)\right)0\right]=L\begin{bmatrix}\rho_{t-1}\otimes\Delta x_{t-1}^{[k-1]}\\ \vdots\\ \rho_{t-\log}\otimes\Delta x_{t-\log}^{[k-1]}\end{bmatrix}=L_t\begin{bmatrix}\Delta x_{t-1}^{[k-1]}\\ \vdots\\ \Delta x_{t-\log}^{[k-1]}\end{bmatrix},$$

[0113] 其中，L_t 是随时间变化的且具有与项L组合在一起的时变调度参数 ρ_t。并且，如果对于所有t，ρ_t 都是有界的，则 L_t 类似地也是有界的。

[0114] 现在，当用块向量的形式将状态方程写成 $X_{1:N}^i=ver\left[x_{1:N}^i\cdots x_1^i\right]$ 时，其中vec操作是将

矩阵的列项(columns)从左边列开始堆栈(stack)在上方，上述结果表明 $\begin{bmatrix}\Delta x_N^{[k]}\\ \vdots\\ \Delta x_{t\text{-}k}^{[k]}\\ \vdots\end{bmatrix}=\begin{bmatrix}0 & L_{N\text{-}1} & \rightarrow & \rightarrow\\ 0 & \ddots & \cdots & \cdots\\ 0 & 0 & L_{N\text{-}k} & \rightarrow\\ 0 & 0 & 0 & \ddots\end{bmatrix}\begin{bmatrix}\Delta x_{tN\text{-}1}^{[k\text{-}1]}\\ \vdots\\ \Delta x_{N\text{-}1}^{[k\text{-}1]}\\ \vdots\end{bmatrix}$，其中→表示$L_{N-k}$延伸到右边。使用M来表示上三角矩阵可以为连续的迭代k和k-1处的状态序列$X_{lag:N}^{i}$之间的差值$\Delta X_{lag:N}^{i}$提供基础表述，如下面的方程36。

[0115] $\Delta X_{lag+2:N}^{i}=M\Delta X_{lag+1:N-1}^{i-1}$ 方程36

[0116] 根据方程36可以看出，LPV情况下迭代的末端收敛速率由M控制，特别是M的最大奇异值。因而M的各种块可通过$(A\text{-}KC)^{i-1}(B\text{-}KD)_{\rho\otimes x}\begin{bmatrix}\rho_{N-1}(1)I_{\dim x}\\ \vdots\\ \rho_{N-i}(\dim p)I_{\dim x}\end{bmatrix}$来计算，其中下标$\rho\otimes x$表示选择B-KD的子矩阵，具有与$\tilde{u}_t$中相应行$\rho\otimes x$对应的列。

[0117] 在一些其它示例性的实施例中，迭代线性子空间计算的收敛性受LPV差分方程稳定性的影响，且更具体地，是受这里描述的LPV线性子空间系统识别的稳定性的影响。因为当且仅当状态转移矩阵的所有的特征值是稳定的，例如，所有特征值都小于1时，一组时不变线性状态空间差分方程才可能是稳定的。LPV情况更复杂，但是为了该示例性的实施例的目的，对于状态向量x_t的每个特征向量的分量来说，每采样时间的增长或收缩的速率可以根据方程9由LPV状态转换矩阵的相应的特征值给出，现在如下面的方程37所示。

[0118] $A(\rho_t)=A_0+\sum_{i=1}^{s}\rho_t(i)A_i$ 方程37

[0119] 在方程37中，假设矩阵$A=(A_0\ A_\rho)=(A_0\ A_1\ \cdots\ A_s)$是未知的常数矩阵。因此，明显的是，转换矩阵$A(\rho_t)$是矩阵$A_i(0\le i\le s)$的线性组合$[1;\rho_t]$。因此，对于矩阵$A_i(0\le i\le s)$的任何选择，存在可能的值$\rho_t$，该值$\rho_t$可在特定的采样时间t产生不稳定的奇异值。如果或当值ρ_t只是零星地产生和/或只是在有限个连续观测时间产生，则不会出现问题或产生误差。

[0120] 在一些其它示例性的实施例中，例如在子空间算法的第k次迭代计算中，只可以使用状态序列$\hat{x}_t^{[k]}$的估计值和矩阵$\hat{A}_i^{[k]}$（$0\le i\le s$）。因此，如果这些数量的估计中的较大误差是允许的或可接受的，则在很大程度上存在不稳定的现象。例如，存在更有意义的重要性的应用领域，如包括飞机机翼颤振动力学识别，其中颤振动力学是边缘稳定的或甚至不稳定的，可以通过从传感器到机翼控制表面的主动控制反馈(active control feedback)以使颤振稳定。然后，在一些其它的应用中，可能保证的是：并不是调度参数ρ_t、矩阵A_i的值和不确定的任何组合都会产生不稳定。进一步地关于这方面，转换矩阵的稳定性可以提供问题的潜在可能，因此需要进一步的估计。

[0121] 在一些其它示例性的实施例中，不稳定性可以产生这样的现象：预测的系统响应可快速地大量增长。例如，当奇异值都是有界的，且都小于1，则预测的系统响应也是有界的，但是如果估计误差很大，例如一段时间的一个奇异值等于2，则预测的响应在该时间内

大约加倍每个采样。因此，如果$10^{30}=(10^3)^{10}=(2^{10})^{10}=2^{100}$，则在如100个这样少的采样中，在计算时会损失30位(digits)的精度，这会产生毫无无意义的结果。

[0122] 因此，根据上述论述，存在这样的情况，在该情况下，与如下时间段(periods)有关的数值精度会产生相当大的损失：在该时间段中，LPV转换矩阵是不稳定的，例如具有延长的时间间隔。并且，困难在于算法初始化和采样大小，因为在具有较大采样的最优解处，该算法是稳定的且收敛。如果用无限的精度进行计算是可能的，则可以避免病态的问题；但是，具有小数点后15或30位的精度，例如，飞机机翼颤振的一些实际数据组，该计算就可以从进一步的估计中获益。

[0123] 因此，一些示例性的实施例可以如下方式进行处理：校正或者减少由算法不稳定导致的任何不需要的效应。例如，当根据先前提到的末端收敛结果时，如果状态序列$\hat{x}_t^{[k]}$充分接近最优值，则该迭代算法是稳定的(如果假设LPV系统是稳定的)。因此，在一些例子中，估计$\hat{x}_t[1]$中大的初始误差会导致不稳定的计算。

[0124] 在其它的例子中，如在时间间隔期间，当调度参数值导致$A(\rho_t)$中存在相当大的不稳定时，$f_t|q_t$的一些项(components)会变得非常大，如被校正用于新值输入q_t产生的效应的新值输出f_t。因此，在$f_t|q_t$的协方差矩阵计算中，这样的大值会导致数值精度存在相当大的病态和损失。

[0125] 在下面的例子中，迭代算法和一些其它子空间方法允许在产生不稳定时任意编辑时间，对不稳定的区域执行异常值(outlier)编辑。例如，在ARX协方差计算中，计算校正的新值，可确定具有相当大的不稳定的时间间隔，并且可以从计算中删除该时间间隔。

[0126] 在处理实验设计的例子中，如果操作点变量的轨线是指定的，则调度参数被安排来以几种方式增强系统识别。选择能避免计算不稳定的初始区域来获得LPV参数(A,B,C,D)的足够精确的初始估计。然后，在其它区域可使用该初始估计对算法的状态估计进行初始化，该算法具有不会出现计算不稳定的足够精确度。

[0127] 在上述例子中，在每次迭代时删除不稳定的异常值是最通常和稳健的步骤。当状态序列$\hat{x}_t^{[k]}$和LPV参数的估计的值随着更多迭代改进时，可以期望减少异常值的数量直到具有快速末端收敛。该期望的反例是，当调度参数ρ_t时程(time history)的开始具有较小的变化，从而该数据这部分的LPV模型对于该数据这部分是有利的，然而却是差的全局模型。然后，在时程的后面部分，ρ_t中可存在估计变化，从而产生不稳定的现象。

[0128] 并且，在这里描述的方法和系统的许多潜在的示例性的应用中，期望提议的算法要比目前工业问题方面不可行的现有方法表现得更好。并且，在许多情况下，期望设计实验来以尽可能少的时间和资源获得操作空间的指定全局区域的期望保真度的结果。由于迭代算法的线性子空间方法是基于极大似然的方法，因此可以改进设计用于LTI系统识别。并且，当用估计的干扰模型确定随机模型时，该干扰模型包括如动态频响函数的置信水平，可以根据干扰过程上的较少的先前信息来推导需要的采样大小和系统输入激励。

[0129] 在一些进一步示例性的实施例中，这里描述的LPV方法和系统可以延伸到非线性系统。例如，示出的是，许多复杂的非线性系统可以用近似LPV的形式表示，该LPV形式充分应用了这里描述的LPV子空间系统识别方法。

[0130] 在一个例子中，通常的非线性、时变、变参数动态系统可以用非线性状态方程的系

统描述，如方程38和39所示的这些方程。

[0131] $x_{t+1}=f(x_t,u_t,\rho_t,v_t)$ 方程38

[0132] $y_t=h(x_t,u_t,\rho_t,v_t)$ 方程39

[0133] 在方程38和39中，x_t是状态向量，u_t是输入向量，y_t是输出向量且v_t是白噪声测量向量。在一些示例性的实施例中，为了处理'变参数'系统，时变参数的'调度'变量ρ_t可以描述系统的当前操作点。非常普遍的函数类f(·)和h(·)可以用增加的(additive)不需要连续的波莱尔函数(borel functions)来表示。

[0134] 以简化的方式，采用如Rugh，部分6.3(W.J.Rugh著，非线性系统原理：Volterra/Wiener方法，由马里兰州巴尔的摩的约翰·霍普金斯大学出版社1981出版，其内容通过引用整体并入本申请)中泰勒展开的函数的情况可以提供如下方程，如方程40和41所示，其中不存在ρ_t和v_t。

[0135] $$x_{t+1}=\sum_{i=0}^{I}\sum_{j=0}^{J}F_{ij}x_t^{(i)}u_t^j$$ 方程40

[0136] $$y_t=\sum_{i=0}^{I}\sum_{j=0}^{J}H_{ij}x_t^{(i)}u_t^j$$ 方程41

[0137] 在方程38和39中，符号$x_t^{(i)}$可以递归地定义为$x_t^{(i)}=x_t\otimes x_t^{(i-1)}$，且类似地定义$u_t^j$。

[0138] 因此，方程40和41是非线性函数f(·)和h(·)的多项式展开。注意到的是，非线性方程可以包括相对简单的形式的非线性函数，如只包括容易计算的用于各种目的积之和的近似多项式方程。然而，对于存在自相关误差时系数的经验估计，对于低维数的y、u和x，问题会变得困难，即使使用了子空间方法。对于子空间方法，矩阵的维数会随着可以使用的'旧值'的维数成指数增长。这发生在展开方程40中，重复将方程40左边的x_t代入状态方程40右边的$x_t^{[k]}$中。因此，用t-1替换t的方程40的整个右边可以提高到幂(power)lagp，即通常选作估计的ARX模型阶的旧值的阶。对于相对低阶的旧值以及x_t和u_t的低维数，增加的项的数量成指数增长。

[0139] 然而，在进一步示例性的实施例和下面的Rugh中，方程39和40可通过卡莱曼双线性转换成双线性向量差分方程，其中状态变量如方程42所示：

[0140] $$x_t^{\otimes}=\left\lfloor x_t^{(1)};x_t^{(2)};\cdots x_t^{(I)}\right\rfloor$$ 方程42

[0141] 且输入幂和乘积变量如方程43所示。

[0142] $$u_t^{\otimes}=\left\lfloor u_t^{(1)};u_t^{(2)};\cdots u_t^{(J)}\right\rfloor$$ 方程43

[0143] 在进一步示例性的实施例中，表示为状态-仿射的形式的方程40可写成如下面的方程44和45所示。

[0144] $$x_t^{\otimes}=A\left(x_t^{\otimes}\otimes u_t^{\otimes}\right)+Bu_t^{\otimes}$$ 方程44

[0145] $$x_t^{\otimes}=C\left(x_t^{\otimes}\otimes u_t^{\otimes}\right)+Du_t^{\otimes}$$ 方程45

[0146] 接下来，假设系数矩阵(A,B,C,D)是调度参数ρ_t的线性函数，表示为(A(ρ_t),B(ρ_t),C(ρ_t)D(ρ_t))且是如方程9至12的状态-仿射的形式。调度参数ρ_t是操作点或者其它已知或精

确测量的变量的非线性函数。例如，由于输入$u_t^{\otimes}$是乘法的且可实时假设为已知的或精确测量的，因此它们可用于调度参数ρ_t中，因而可能会减少它们的维数。双线性方程变成下面的方程46和47所示。

[0147] $$x_t^{\otimes}=A(\rho_t)x_t^{\otimes}+B(\rho_t)u_t^{\otimes}$$ 方程46

[0148] $$x_t^{\otimes}=C(\rho_t)x_t^{\otimes}+D(\rho_t)u_t^{\otimes}$$ 方程47

[0149] 方程46和47是明显的LPV形式。因此，对于充分平滑的函数$f(x_t,u_u,\rho_t,v_t)$和$h(x_t,u_u,\rho_t,v_t)$，这些函数在变量附近(neighborhood)存在明确定义的幂级数展开，可存在过程的LPV近似。

[0150] 因此，将输入$u_t^{\otimes}$代入调度参数ρ_t中，方程可以是线性时变的且在一些例子中可以具有更高的维数ρ_t。这可以显示输入和调度参数的双功能，以及它们如何以各种方式相互交换，例如，它们的差值是随时间变化的速率(rate)。因此调度的知识可以从基础物理、化学或其它基础知识中获得。因此，当灰匣子(graybox)模型在如下范围内：它们能够结合关于过程的行为的相当大的全局信息，该过程可以并入过程动态如何随操作点改变的模型中时，有时可以估计LPV模型。

[0151] 前面的描述和附图阐述了本发明的原理、优选的实施例和操作模式。然而，本发明不应认为限于上面描述的特定实施例。本领域技术人员将会知道上述实施例的另外的变形。

[0152] 因此，上述实施例应该被认为是例证性的而不是限制性的。因此，应该理解的是，本领域技术人员可以对这些实施例进行变形，只要其不背离下面的权利要求限定的本发明的范围。

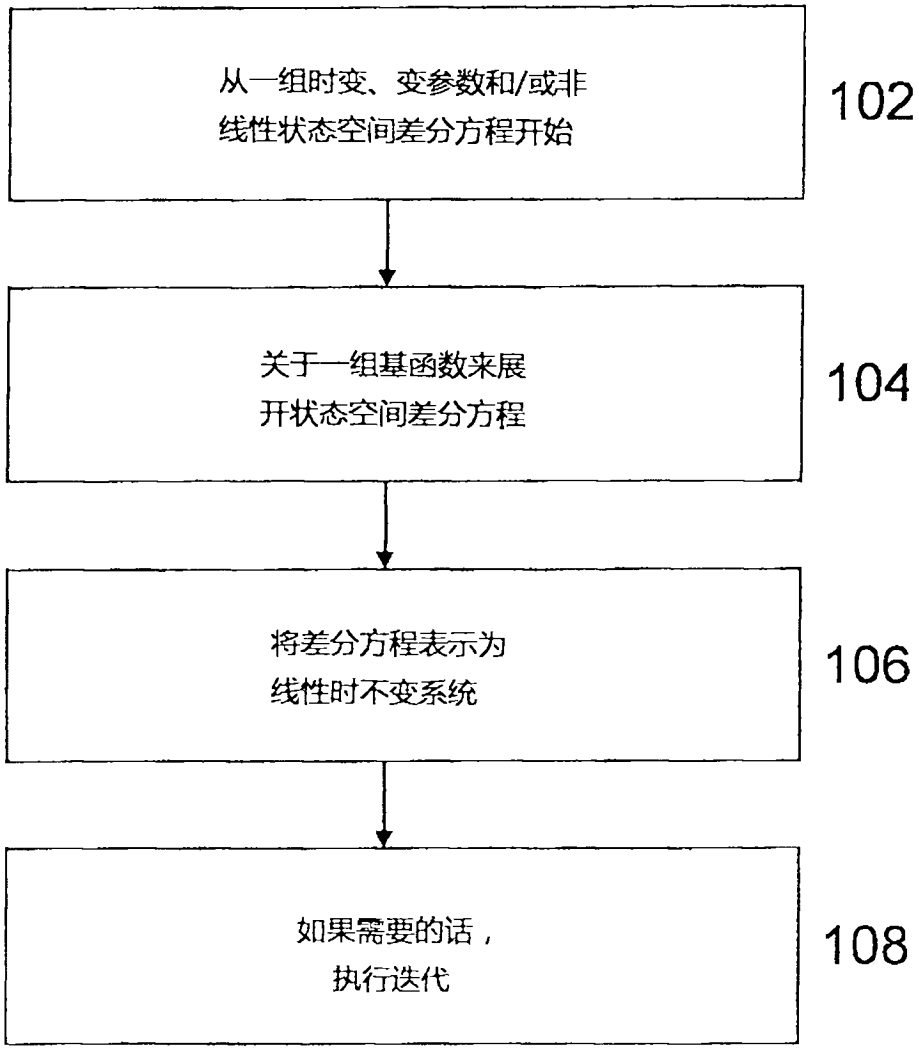

从一组时变、变参数和/或非
线性状态空间差分方程开始
102
关于一组基函数来展
开状态空间差分方程
104
将差分方程表示为
线性时不变系统
106
如果需要的话，
执行迭代
108

图1

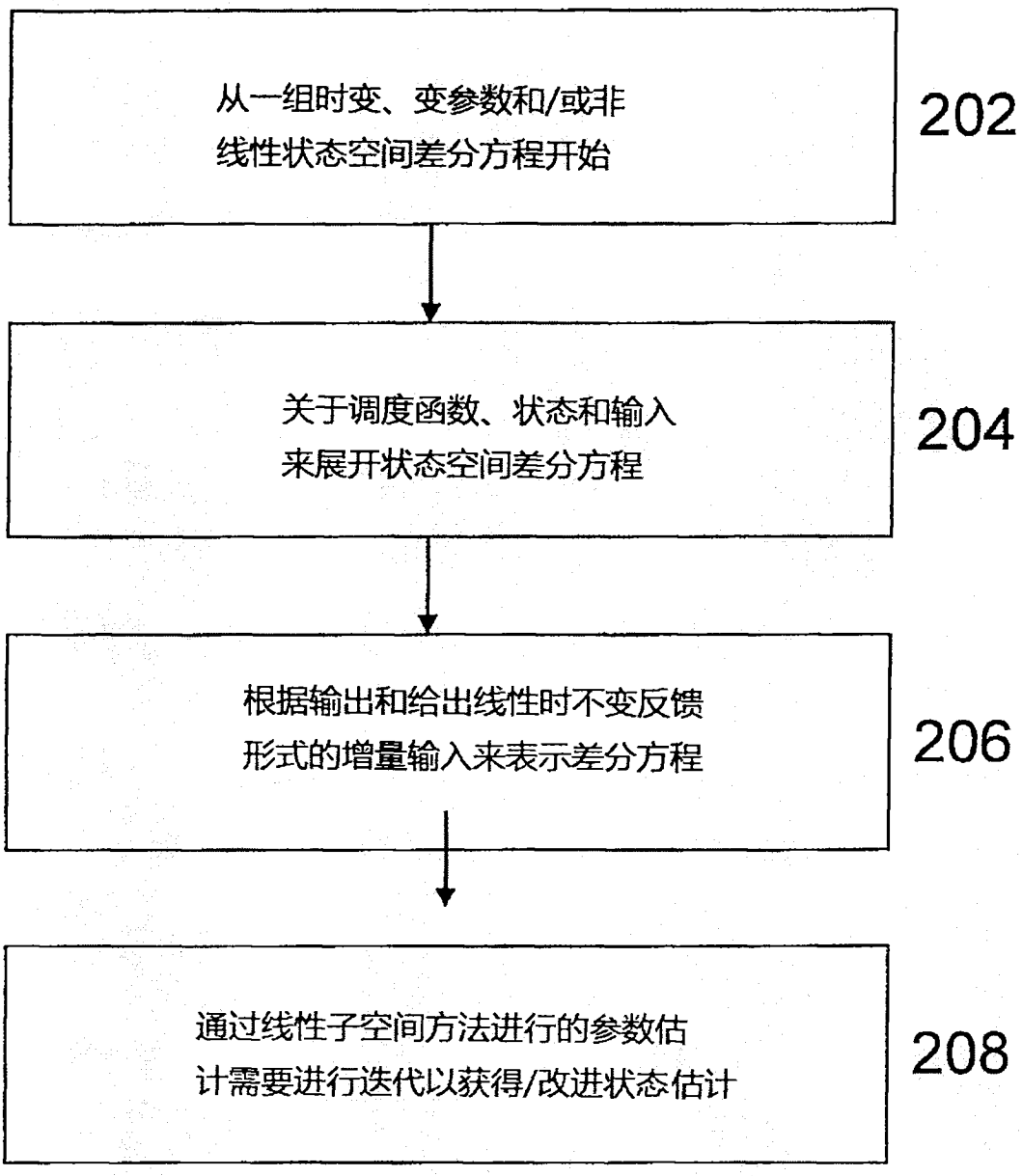

从一组时变、变参数和/或非
线性状态空间差分方程开始
202
关于调度函数、状态和输入
来展开状态空间差分方程
204
根据输出和给出线性时不变反馈
形式的增量输入来表示差分方程
206
通过线性子空间方法进行的参数估
计需要进行迭代以获得/改进状态估计
208

图2

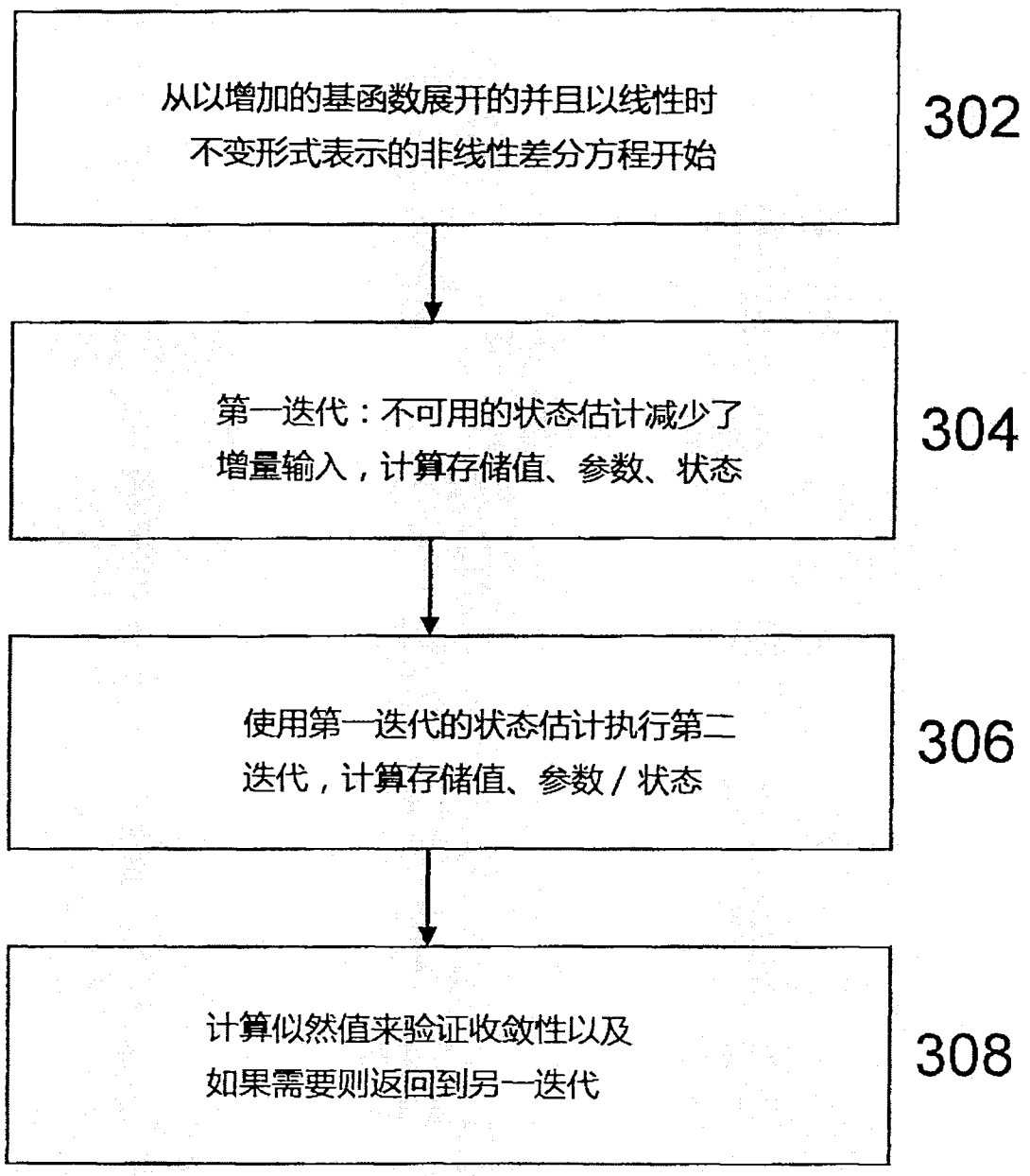

从以增加的基函数展开的并且以线性时
不变形式表示的非线性差分方程开始
302
第一迭代：不可用的状态估计减少了
增量输入，计算存储值、参数、状态
304
使用第一迭代的状态估计执行第二
迭代，计算存储值、参数 / 状态
306
计算似然值来验证收敛性以及
如果需要则返回到另一迭代
308

图3

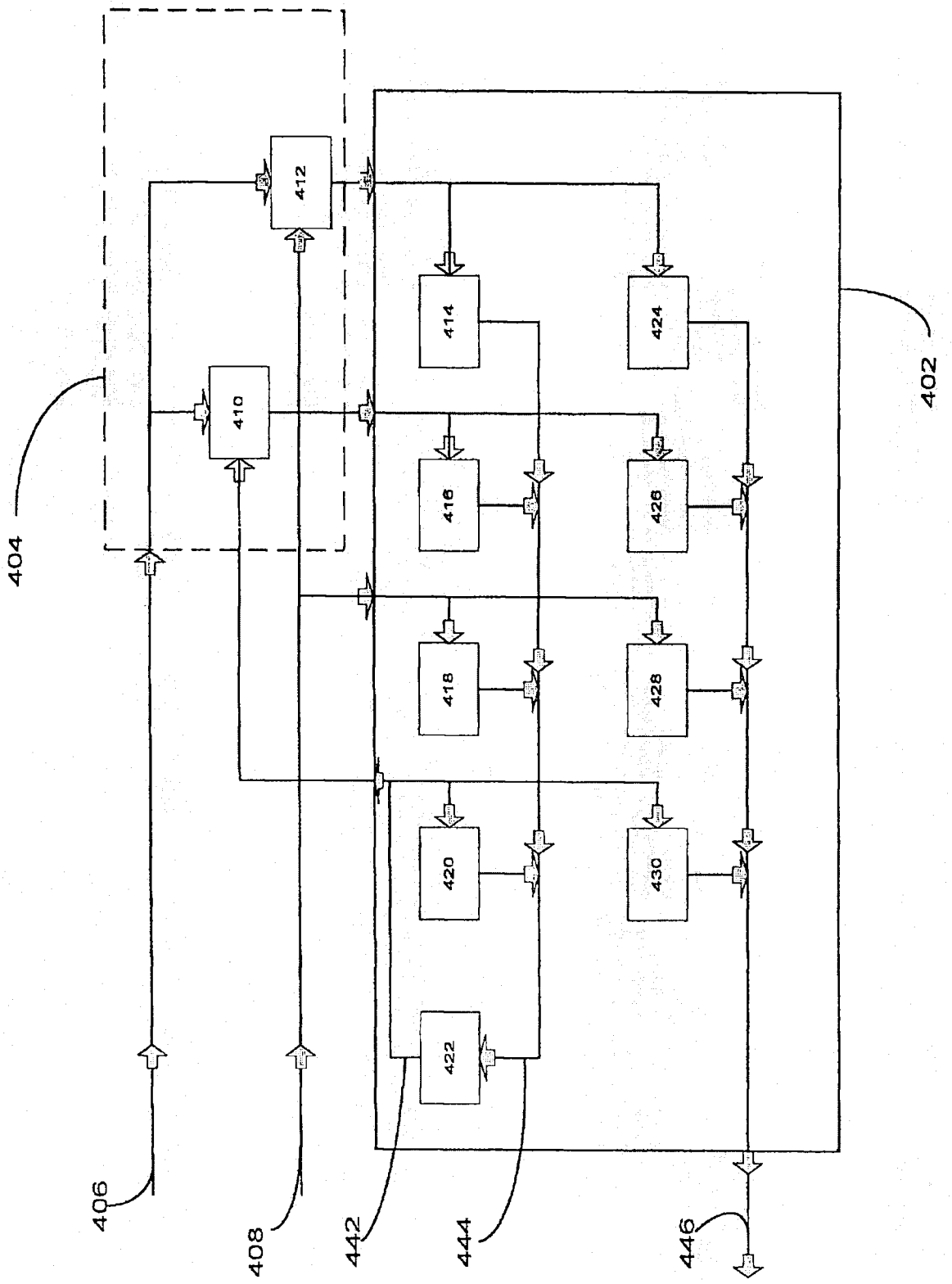

412
414
424
410
416
426
418
428
420
430
422
402
404
406
408
442
444
446

图4

502 504 506

参量	迭代 1	迭代 k	迭代 k+1
期望步骤：			
输出向量	y_t	y_t	y_t
输入向量 $\tilde{u}_t$	$\begin{bmatrix} u_t \\ \rho_t \otimes u_t \end{bmatrix}$	$\begin{bmatrix} u_t \\ \rho_t \otimes \hat{x}_t^{[k-1]} \\ \rho_t \otimes u_t \end{bmatrix}$	$\begin{bmatrix} u_t \\ \rho_t \otimes \hat{x}_t^{[k]} \\ \rho_t \otimes u_t \end{bmatrix}$
存储值估计	$\hat{m}_t^{[1]}$	$\hat{m}_t^{[k]}$	$\hat{m}_t^{[k+1]}$
极大值步骤：			
参数估计	$\hat{\Theta}^{[1]}$	$\hat{\Theta}^{[k]}$	$\hat{\Theta}^{[k+1]}$
状态估计	$\hat{x}_t^{[1]}$	$\hat{x}_t^{[k]}$	$\hat{x}_t^{[k+1]}$

图5

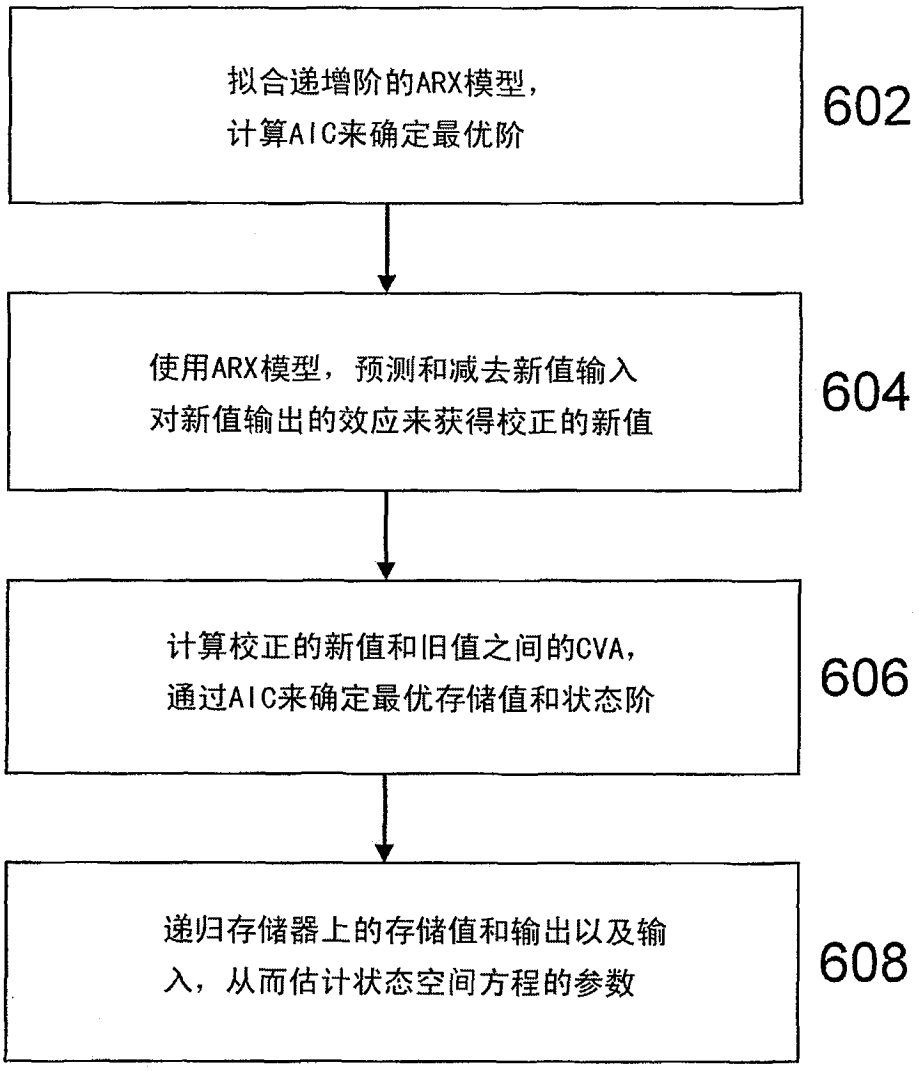

拟合递增阶的ARX模型，
计算AIC来确定最优阶
602
使用ARX模型，预测和减去新值输入
对新值输出的效应来获得校正的新值
604
计算校正的新值和旧值之间的CVA，
通过AIC来确定最优存储值和状态阶
606
递归存储器上的存储值和输出以及输
入，从而估计状态空间方程的参数
608

图6